

CLAIMS

- 1 1. An apparatus comprising:
2 at least one processor;
3 a memory coupled to the at least one processor;
4 a database residing in the memory, the database supporting batch updates;
5 generated code residing in the memory and executed by the at least one processor,
6 the generated code interacting with the database using a plurality of calls to the database;
7 and
8 a batch mechanism residing in the memory and executed by the at least one
9 processor, the batch mechanism processing the plurality of calls by the generated code
10 and batching a plurality of database updates in the plurality of calls by the generated code
11 to the database.
- 1 2. The apparatus of claim 1 wherein the batch mechanism intercepts at least one call
2 to the database from the generated code.
- 1 3. The apparatus of claim 2 wherein the at least one call includes a call to prepare a
2 statement.
- 1 4. The apparatus of claim 2 wherein the at least one call includes an executeUpdate()
2 call.
- 1 5. The apparatus of claim 2 wherein the at least one call includes a
2 returnPreparedStatement() call.

- 1 6. The apparatus of claim 1 wherein the batch mechanism creates a batch when a call
2 corresponding to a first update is made by the generated code, adds an update to the batch
3 for each subsequent call corresponding to an update that is not a last call, and executes
4 the batch when the generated code makes the last call corresponding to a last update.
- 1 7. The apparatus of claim 1 wherein the batch mechanism batches the plurality of
2 updates without affecting the generated code.

- 1 8. An apparatus comprising:
2 at least one processor;
3 a memory coupled to the at least one processor;
4 a database residing in the memory, the database supporting batch updates;
5 generated code residing in the memory and executed by the at least one processor,
6 the generated code interacting with the database using a plurality of calls to the database,
7 the plurality of calls including a plurality of calls to a prepareStatement() method, a
8 plurality of calls to an executeUpdate() method, and a plurality of calls to a
9 returnPreparedStatement() method; and
10 a batch mechanism residing in the memory and executed by the at least one
11 processor, the batch mechanism creating a batch when an executeUpdate() call
12 corresponding to a first update is made by the generated code, adding an update to the
13 batch for each subsequent executeUpdate() call corresponding to an update that is not a
14 last call, and executing the batch when the generated code makes the last executeUpdate()
15 call corresponding to a last update.
- 1 9. The apparatus of claim 8 wherein the batch mechanism closes and caches a
2 prepared statement corresponding to a call when the batch mechanism is done executing
3 the batch.
- 1 10. The apparatus of claim 8 wherein the batch mechanism does not affect the
2 generated code.

- 1 11. A computer-implemented method for enabling batch processing of database
2 updates without affecting generated code that executes calls to a database that supports
3 batch operations, the method comprising the steps of:
4 providing a batch mechanism that intercepts a plurality of calls from the generated
5 code to the database; and
6 the batch mechanism batching a plurality of database updates in the plurality of
7 calls by the generated code to the database, and executing the batch to the database.
- 1 12. The method of claim 11 wherein the plurality of calls from the generated code
2 intercepted by the batch mechanism include a call to prepare a statement.
- 1 13. The method of claim 11 wherein the plurality of calls from the generated code
2 intercepted by the batch mechanism include a plurality of calls to an executeUpdate()
3 method.
- 1 14. The method of claim 11 wherein the plurality of calls from the generated code
2 intercepted by the batch mechanism include at least one call to a
3 returnPreparedStatement() method.
- 1 15. The method of claim 11 further comprising the steps of:
2 creating a batch when a call corresponding to a first update is made by the
3 generated code;
4 adding an update to the batch for each subsequent call corresponding to an update
5 that is not a last call; and
6 executing the batch when the generated code makes the last call corresponding to
7 a last update.

1 16. A computer-implemented method for enabling batch processing of database
2 updates without affecting generated code that executes calls to a database that supports
3 batch operations, the method comprising the steps of:
4 providing a batch mechanism that intercepts the calls from the generated code to
5 the database, the calls including a plurality of calls to a prepareStatement() method, a
6 plurality of calls to an executeUpdate() method, and at least one call to a
7 returnPreparedStatement() method;
8 creating a batch when a call corresponding to a first update is made by the
9 generated code;
10 adding an update to the batch for each subsequent call corresponding to an update
11 that is not a last call; and
12 executing the batch when the generated code makes the last call corresponding to
13 a last update.

- 1 17. A program product comprising:
2 a batch mechanism that processes a plurality of calls by generated code that
3 interacts with a database using a plurality of calls to the database, the batch mechanism
4 batching a plurality of database updates in the plurality of calls by the generated code to
5 the database; and
6 computer readable signal bearing media bearing the batch mechanism.
- 1 18. The program product of claim 17 wherein the signal bearing media comprises
2 recordable media.
- 1 19. The program product of claim 17 wherein the signal bearing media comprises
2 transmission media.
- 1 20. The program product of claim 17 wherein the batch mechanism intercepts at least
2 one call to the database from the generated code.
- 1 21. The program product of claim 20 wherein the at least one call includes a call to
2 prepare a statement.
- 1 22. The program product of claim 20 wherein the at least one call includes an
2 executeUpdate() call.
- 1 23. The program product of claim 20 wherein the at least one call includes a
2 returnPreparedStatement() call.

1 24. The program product of claim 17 wherein the batch mechanism creates a batch
2 when a call corresponding to a first update is made by the generated code, adds an update
3 to the batch for each subsequent call corresponding to an update that is not a last call, and
4 executes the batch when the generated code makes the last call corresponding to a last
5 update.

1 25. The program product of claim 17 wherein the batch mechanism batches the
2 plurality of updates without affecting the generated code.

- 1 26. A program product comprising:
2 a batch mechanism that creates a batch when an executeUpdate() call
3 corresponding to a first update is made by generated code that interacts with a database
4 using a plurality of calls to the database, the plurality of calls including a plurality of calls
5 to a prepareStatement() method, a plurality of calls to an executeUpdate() method, and a
6 plurality of calls to a returnPreparedStatement() method, the batch mechanism adding an
7 update to the batch for each subsequent executeUpdate() call corresponding to an update
8 that is not a last call, and executing the batch when the generated code makes the last
9 executeUpdate() call corresponding to a last update; and
10 computer readable signal bearing media bearing the batch mechanism.
- 1 27. The program product of claim 26 wherein the signal bearing media comprises
2 recordable media.
- 1 28. The program product of claim 26 wherein the signal bearing media comprises
2 transmission media.
- 1 29. The program product of claim 26 wherein the batch mechanism closes and caches
2 a prepared statement corresponding to a call when the batch mechanism is done executing
3 the batch.
- 1 30. The program product of claim 26 wherein the batch mechanism does not affect the
2 generated code.

* * * * *